Great question, Gyanenjay! Let's break down \*\*future.get()\*\* and when/how to use it:

**✅ What is future.get()?**

* It's a **blocking call**.
* It waits for the task to finish and **returns the result** produced by a Callable.

**🔁 How it works:**

Future<String> future = executor.submit(() -> {

Thread.sleep(3000); // Simulate long task

return "Completed";

});

String result = future.get(); // waits here until result is ready

System.out.println(result); // prints "Completed"

**📌 Key Points:**

| **Behavior** | **Description** |
| --- | --- |
| Blocking | get() blocks the thread until the task finishes. |
| Result Retrieval | It gives the return value from the Callable. |
| Exception | If the task throws an exception, get() will throw an ExecutionException. |

**⏳ Use with Timeout (Recommended for safety):**

future.get(5, TimeUnit.SECONDS); // Wait max 5 seconds

Throws TimeoutException if not completed in time.

**❗️Be Careful:**

If your task takes too long, .get() will make your main thread **wait**, which could make your app slow.

**🧠 Pro Tip for Interviews:**

* Mention future.get() is **synchronous** (blocks).
* For **non-blocking**, prefer CompletableFuture.

Want me to show an example where a task takes too long and we use get(timeout) to avoid waiting forever?